

# Jeu sérieux et motivation des étudiants pour apprendre : influence du contexte avec Prog&Play

► Mathieu MURATET (INS HEA, Suresnes), Elisabeth DELOZANNE (LIP6, Paris), Patrice TORGUET (IRIT, Toulouse), Fabienne VIALLET (UMR EFTS, Toulouse)

---

---

■ **RÉSUMÉ** • Cet article présente l'analyse de l'utilisation à grande échelle de Prog&Play (Prog&Play, 2012), un jeu sérieux spécialement conçu pour l'enseignement des fondamentaux de la programmation aux étudiants en première année universitaire. L'étude s'appuie principalement sur les résultats d'une enquête sur la motivation des étudiants. Cette enquête a été réalisée auprès de 182 étudiants parmi 258 qui ont utilisé le jeu sérieux entre 4 et 20 heures dans sept contextes universitaires différents. Nos résultats montrent que l'intérêt des étudiants pour Prog&Play ne dépend pas seulement des qualités intrinsèques du jeu, mais dépend aussi du contexte d'enseignement et principalement de la manière dont les enseignants organisent les séances pour exploiter la technologie.

■ **MOTS-CLÉS** • Jeux sérieux, programmation, algorithmique, motivation.

■ **ABSTRACT** • *This paper deals with an analysis of a large-scale use of Prog&Play (Prog&Play, 2012), a game-based learning environment specially designed to teach the basics of programming to first year university students. The study relies mainly on a motivation survey completed by 182 students among 258 who used the serious game for 4 to 20 hours in seven different university settings. Our findings show that the students' interest for Prog&Play is not only related to the intrinsic game quality, it is also related to the teaching context and mainly to the course schedule and the way teachers organize sessions to benefit from the technology.*

■ **KEYWORDS** • *Serious games, programming, algorithms, motivation*

## 1. Introduction

De nombreuses études mentionnent, dans les pays développés, un désintéressement grandissant des étudiants pour les sciences en général et pour l'informatique en particulier (CSC, 2008) et (Papastergiou, 2009). Pour répondre au besoin urgent de formation d'informaticiens, plusieurs pays mettent en place des programmes d'enseignement de l'informatique (PSD, 2012), (Archambault, 2011) et (Seehorn *et al.*, 2011). Parallèlement, des recherches se développent pour concevoir des approches pédagogiques afin de rendre les formations d'informatique plus accessibles et attractives pour les étudiants. Ces approches incluent des compétitions internationales entre écoles (Bebras, 2012) ou entre pays (IOI, 2012) et (Février *et al.*, 2011). D'autres études montrent que le jeu vidéo est un moyen efficace pour stimuler la motivation des étudiants en rendant l'apprentissage plus ludique. Par exemple, pour Oblinger (Oblinger, 2004), les jeux vidéo supportent des apprentissages fondés sur la résolution de problème ou l'expérimentation car ils fournissent des retours immédiats en permettant au joueur d'autoévaluer leurs actions et stratégies de jeu. Le travail présenté dans cet article est une contribution à ce domaine de recherche. Trois séries d'enquêtes réalisées de septembre 2007 à juin 2010 auprès de 900 étudiants répartis dans huit formations différentes (Muratet, 2010) ont montré une réelle appétence des étudiants pour les jeux vidéo et notamment les jeux de stratégie temps réel qui peuvent constituer un contexte intéressant pour l'apprentissage de la programmation.

Le projet Prog&Play a ainsi comme objectif d'accroître la motivation des étudiants pour l'apprentissage des fondamentaux de la programmation. Pour cela, nous avons conçu et réalisé un logiciel aussi nommé Prog&Play qui propose aux étudiants d'écrire des programmes qui contrôlent les unités d'un jeu de stratégie temps réel. S'ils programment une stratégie efficace, ils augmentent leurs chances de battre leurs adversaires et de remporter des victoires. Dans un article précédent, nous avons détaillé la conception, la réalisation et l'évaluation de Prog&Play (Muratet *et al.*, 2011). Dans cet article, nous étudions l'influence du contexte d'enseignement, dans lequel s'intègre Prog&Play, sur la motivation des étudiants. Dans un premier temps, nous situons nos travaux par rapport à différentes recherches publiées sur l'utilisation des jeux sérieux pour l'apprentissage de la programmation. Dans un deuxième temps, nous décrivons le jeu sérieux Prog&Play avant de présenter et d'analyser les données que nous avons recueillies lors de différentes expérimentations conduites auprès de 258 étudiants et de 20 enseignants dans différents contextes universitaires. Pour terminer, nous analysons les résultats afin de proposer des recommandations pour une utilisation efficace d'une approche ludique de l'apprentissage de la programmation.

## 2. Travaux connexes

Les jeux vidéo sont utilisés pour enseigner la programmation de deux façons différentes. La première consiste à demander aux étudiants de concevoir et réaliser leur propre jeu vidéo. Par exemple, Chen et Cheng (Chen et Cheng, 2007) demandent aux étudiants, à travers un projet collaboratif, d'implémenter en C++ un mini jeu vidéo interactif. Des environnements de programmation pour débutants comme Scratch (Maloney *et al.*, 2004), Alice2 (Kelleher *et al.*, 2002) ou StarLogo TNG (Klopfer et Yoon, 2005) peuvent être utilisés en complément pour faciliter les premières expériences de programmation.

Prog&Play se situe dans une seconde approche où le joueur écrit des programmes pour contrôler les actions des unités d'un jeu vidéo. Certains projets favorisent la motivation en s'appuyant sur une narration. Par exemple, dans Colobot (Colobot, 2001), le joueur colonise une planète en utilisant des robots qu'il programme avec un langage de programmation similaire au C++. D'autres projets misent sur la compétition pour motiver les joueurs. Dans Robocode (Hartness, 2004), le but est de programmer un char d'assaut pour combattre ceux des autres joueurs. D'autres exemples de ce type de jeu existent comme Gun-Tactyx (Gun-Tactyx, 2008) qui utilise le langage SMALL ou Robot Battle (Robot Battle, 2009) qui utilise un langage de script spécialement conçu pour le jeu.

## 3. Présentation de Prog&Play

Prog&Play est basé sur trois principes : (i) l'engagement des apprenants dans le jeu est suscité soit par une narration soit par la compétition entre les joueurs ; (ii) les apprenants programment les unités du jeu avec des programmes simples impliquant l'utilisation d'une bibliothèque de fonctions personnalisable par les enseignants ; (iii) les apprenants voient les résultats de leurs programmes dans le contexte du jeu.

Prog&Play dispose d'une interface de programmation, disponible en différents langages, pour programmer les unités du jeu de stratégie temps réel *Kernel Panic* (Kernel Panic, 2012). *Kernel Panic* est un jeu multijoueur en 3D au code source libre dont l'univers se positionne au sein même d'un ordinateur. Dans ce contexte, le joueur prend le contrôle des différentes unités du jeu comme par exemple des Bits, des Octets et des Assembleurs. Dans le projet Prog&Play, deux modes de jeu sont possibles. Dans le premier, la motivation s'appuie sur une dimension narrative (les étudiants ont à réussir des missions comme dans Colobot), dans le second, l'aspect multijoueur de *Kernel Panic* est exploité pour introduire une dimension compétitive. Dans le cadre de cet article, nous nous concentrons sur le premier mode de jeu.

L'approche narrative encapsule les objectifs pédagogiques dans les différentes missions qui sont à effectuer. Le scénario exploite l'univers de *Kernel Panic* : « De-

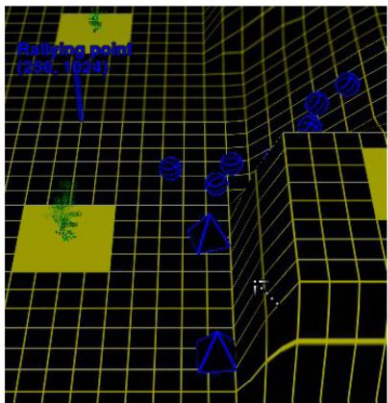
*puis un certain nombre d'années, une guerre secrète fait rage au sein même des ordinateurs. Des attaques ont régulièrement lieu contre d'innocentes victimes. Aujourd'hui c'est votre tour. Votre agresseur a capturé le contrôleur de votre souris. Vous devez le récupérer. Votre seule solution : la programmation ».* Une séquence de plusieurs missions est proposée aux étudiants en vue d'atteindre l'objectif final. En voici deux exemples :

– Mission 1 : *« Lors de la précédente attaque, vous avez perdu de nombreuses unités et celles qui vous restent sont dispersées sur la carte. Un seul Bit est encore sous votre contrôle. Un rapport vous est parvenu vous indiquant qu'un Octet se trouverait au point de ralliement (1056, 1792). Déplacez votre unique unité à cette position pour tenter de retrouver l'Octet perdu. Bon courage commandant... »* Pour réussir cette mission, le joueur doit écrire un petit programme manipulant des variables, des types, des affectations, des fonctions et le passage de paramètre. L'objectif pédagogique de cette mission est de permettre aux étudiants de prendre contact avec les structures de données et les fonctions de base de la bibliothèque Prog&Play.

– Mission 4 : *« Toutes les unités que vous possédez ont subi de lourds dommages lors de la précédente attaque. Vous devez les réparer avant de lancer la contre-attaque. Le dernier Assembleur encore en marche et capable d'effectuer les réparations se trouve aux coordonnées (256, 1024). Déplacez toute votre armée à la rencontre de cet Assembleur ».* Dans cette mission, l'étudiant doit utiliser une structure de contrôle itérative, pour parcourir l'ensemble des unités et les déplacer à la bonne position. La figure 1 présente une capture d'écran de cette mission (1a) et la solution écrite en langage C (1b) et en Scratch (1c).

Nous avons conçu les missions de façon à permettre aux étudiants d'aborder progressivement les difficultés du jeu. Les premières missions servent à les guider dans les manipulations de base (déplacer, regrouper, réparer des unités) afin de leur permettre de laisser libre cours à leur créativité dans la mission finale. Pour cette dernière, ils imaginent et mettent en œuvre une palette de stratégies et testent immédiatement leurs effets pour les corriger et les affiner.

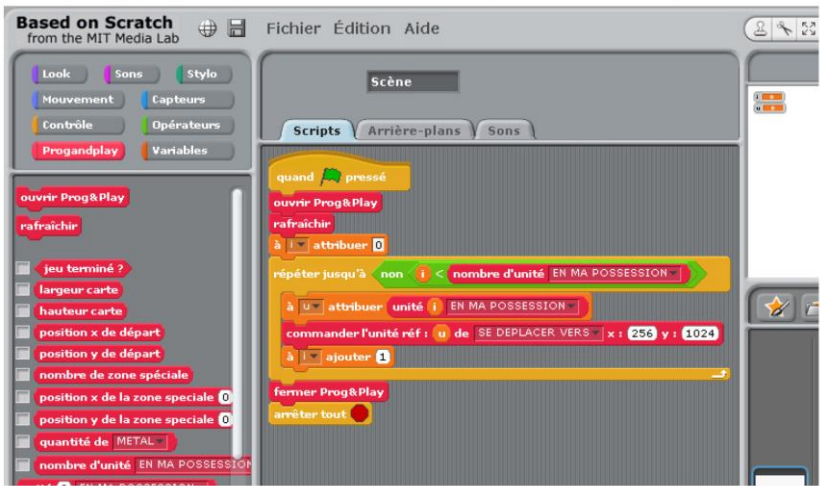
Enfin, pour faciliter l'intégration de Prog&Play dans différents contextes d'enseignement, nous avons développé des interfaces à la bibliothèque dans les six langages de programmation très utilisés pour l'initiation à la programmation : Ada, C/C++, Compalgo, Java, OCaml et Scratch.



(a) Vue du jeu

```
01 - #include "PP_Client.h"
02 - #include "constantList_KP4.1.h"
03 -
04 - int main ()
05 - {
06 -     int i;
07 -     PP_Unit u;
08 -     PP_Pos targetPos;
09 -
10 -     targetPos.x = 256.0;
11 -     targetPos.y = 1024.0;
12 -     PP_Open(); /* Open the game API */
13 -     PP_Refresh(); /* Refresh game state */
14 -     for (i=0 ; i<PP_GetNumUnits(MY_COALITION) ; i++){
15 -         /* Get current unit */
16 -         u = PP_GetUnitAt(MY_COALITION, i);
17 -         /* Order the unit to move to the position */
18 -         PP_Unit_ActionOnPosition(u, MOVE, targetPos);
19 -     }
20 -     PP_Close(); /* Close the game API */
21 - }
```

(b) Solution en langage C



(c) Solution en langage Scratch

Figure 1 • Quatrième mission de Prog&Play : notion d'itération.

4. Évaluation

L'objectif de Prog&Play est d'améliorer la motivation des étudiants. Pour évaluer Prog&Play, nous avons adopté une méthode de conception collaborative et itérative de l'évaluation. Cette méthode implique les enseignants afin d'étudier comment ils mettent en œuvre Prog&Play dans les enseignements de programmation dont ils sont responsables. Nous avons aussi conçu et fait remplir un questionnaire aux étudiants après l'utilisation de Prog&Play. Dans l'étude présentée ici, notre question de recherche est la suivante : Existe-t-il une relation entre la motivation des étudiants et les

contextes d'enseignement dans lesquels Prog&Play a été intégré et quel contexte est le plus pertinent ?

#### 4.1. Contextes d'utilisation

Nous avons étudié l'usage de Prog&Play dans sept contextes différents (notés C1 à C7) qui ont impliqué 258 étudiants et 20 enseignants. Nous avons cherché des contextes différents pour tester la robustesse de l'approche proposée. Les enseignants ont organisé les séances et l'évaluation du travail des étudiants en fonction de leurs contraintes institutionnelles suite à notre sollicitation dans les contextes C1 à C5 et de leur propre initiative dans les contextes C6 et C7. Aucun membre de l'équipe de conception de Prog&Play n'a participé aux enseignements dans les contextes C4, C6 et C7. Dans les contextes C4 et C5, les travaux pratiques avec Prog&Play étaient obligatoires et intégrés dans les enseignements en remplacement des sujets habituels, alors que pour les autres contextes, le jeu était utilisé en complément des enseignements obligatoires. Pour les contextes C6 et C7 les deux enseignants, de leur propre initiative, ont conçu une formation autour de Prog&Play et ceci dans deux universités différentes afin d'expérimenter de nouvelles approches pédagogiques. Ces formations sont reconduites chaque année depuis quatre ans.

Dans chaque contexte, Prog&Play était préinstallé sur les ordinateurs de l'établissement de formation avant l'arrivée des étudiants. Les séances se déroulaient en présence d'au moins un enseignant qui présentait les spécificités du jeu, l'environnement de programmation, la bibliothèque de fonction et répondait aux sollicitations des étudiants. Dans le contexte C6 seulement, après 5 séances en présence d'un enseignant, les étudiants ont terminé le jeu en autonomie avec un support par courriel de l'enseignant et de leurs pairs.

En résumé, les contextes concernent tous l'initiation à un langage de programmation mais le *langage* varie (algorithmique C1, impératif/objet C2, C3, C4, C6, C7, fonctionnel C5). L'*intégration* du jeu dans l'enseignement varie : le jeu est utilisé en remplacement de séances de TP ordinaires sans changement dans le reste du dispositif (en particulier de l'évaluation), en complément d'un enseignement ou dans un enseignement spécialement créé autour du jeu. Le *public* est constitué d'étudiants de première année d'IUT d'informatique ou de licence d'informatique sauf dans le contexte C3. Le public est désigné dans tous les contextes sauf les contextes C1, C2 et C6 où les étudiants étaient volontaires. Enfin le temps alloué à l'utilisation de Pro&Play varie de 4h 30 à 12h ou plus.

#### 4.2. Matériel

Pour récolter les données sur la motivation des étudiants, nous avons conçu un questionnaire (la version actuelle du questionnaire est disponible à l'adresse :

<http://goo.gl/qj3fE>) à partir de la hiérarchie des besoins des joueurs de jeux vidéo proposée par Siang et Rao (Siang et Rao, 2003) et Greitzer (Greitzer *et al.*, 2007). Ces auteurs ont adapté la pyramide des besoins de Maslow pour définir sept critères à satisfaire pour motiver les joueurs dans un jeu vidéo : (besoin 1) le besoin de règles, les joueurs recherchent des informations pour comprendre les règles de base structurant le jeu ; (besoin 2) le besoin de sécurité, les joueurs ont besoin de trouver de l'aide sur le fonctionnement du jeu ; (besoin 3) le besoin d'appartenance, les joueurs ont besoin de s'approprier le jeu pour se sentir capable d'atteindre les objectifs ; (besoin 4) le besoin d'estime, les joueurs ont besoin d'être valorisés par le jeu (rétroactions, progression, score, compétition, etc.) ; (besoin 5) le besoin de connaître et de comprendre, les joueurs ont besoin de découvrir les informations/bonus cachés et de s'en forger une image cohérente en vue de les réinvestir en situation de jeu ; (besoin 6) le besoin d'esthétique, les joueurs ont besoin de graphismes attrayants, d'effets visuels, d'une musique appropriée, d'effets sonores, etc. ; (besoin 7) le besoin d'auto accomplissement, les joueurs veulent être capables de projeter leur créativité et imagination dans le jeu sous contrainte du respect des règles. En suivant ces auteurs, nous supposons que le degré de satisfaction de ces critères est un indicateur significatif de la motivation. Voici quelques exemples des items de notre questionnaire : « *Avez-vous trouvé l'aide suffisante pour comprendre le fonctionnement du jeu ?* » (niveau 2) ; « *Vous êtes-vous senti capable de réaliser les objectifs de chaque mission ?* » (niveau 3) ; « *Appréciez-vous l'esthétique du jeu (univers, effets visuels, etc.) ?* » (niveau 6). Chaque réponse est donnée avec une échelle de Likert à 7 items allant de « Pas du tout = 1 » à « Tout à fait = 7 ».

### 4.3. Résultats et analyse

Nous comparons (Tableau 1) la satisfaction des étudiants pour chaque contexte d'utilisation. Pour les contextes C4 et C5, où les travaux pratiques de Prog&Play étaient obligatoires en remplacement d'un TP usuel, seul un quart des étudiants étaient satisfaits. Dans les contextes C1, C2 et C3 où Prog&Play était utilisé en complément de l'enseignement usuel (comme atelier ou travaux pratiques pour des étudiants en difficulté), le taux d'étudiants satisfaits était de 4 sur 10. Enfin dans le contexte C6 où Prog&Play était utilisé comme un projet, le taux a atteint 6 sur 10.

Ces résultats suggèrent que Prog&Play est mieux adapté sous une forme de projet et de soutien ou d'atelier en complément des formations usuelles. Nous émettons l'hypothèse que Prog&Play n'est pas un jeu qui *enseigne* les fondamentaux de la programmation, mais c'est un micro-monde au sens de Papert (Papert, 1980) où les étudiants peuvent explorer les effets de leurs différents programmes. Ce type d'environnement, favorise un apprentissage par expérimentation où les retours, fournis par le micro-monde, permettent aux étudiants d'autoévaluer leurs tentatives. Les

étudiants utilisent leurs connaissances en programmation dans un contexte ludique qui leur est familier (ici un jeu de stratégie temps réel) alors que, dans les enseignements classiques, ils doivent les mobiliser souvent dans un contexte mathématique abstrait et dans lequel beaucoup ne sont pas à l'aise.

	n/N*	Langage, contexte d'enseignement et temps de jeu	TEG**
C1	13/15	Compalgo, IUT (département Informatique), semestre 1, atelier de programmation en complément de l'enseignement usuel, 5*1h30	4.6/10
C2	23/35	C, IUT (département Informatique), semestre 1, soutien pour des étudiants en difficulté en complément de l'enseignement usuel, 3*1h30	3.8/10
C3	16/16	Java, IUT (département Informatique), semestre 3, atelier de programmation en complément de la formation initiale, 3*1h30	4.1/10
C4	29/60	C, IUT (département Service et Réseaux de Communication), semestre 1, travaux pratiques obligatoires pour tous les étudiants, 5*1h30	2.7/10
C5	91/99	OCaml, université (parcours Informatique, Mathématiques, Physique), semestre 1, travaux pratiques obligatoires pour tous les étudiants, 2*2h	2.6/10
C6	10/18	C, université (unité d'enseignement orientée TICE transverse au L1), semestre 2, projet de programmation en complément d'une initiation au langage C, 6*2h + Travail en autonomie	6.3/10
C7	0/15	C, IUT (département Génie Electrique et Informatique Industrielle), semestre 3, travaux pratiques intégrés à une formation spécialement conçue pour des étudiants en difficulté, 5*2h	Non renseigné

\* n/N : nombre de questionnaires complétés / nombre d'étudiants ayant joué

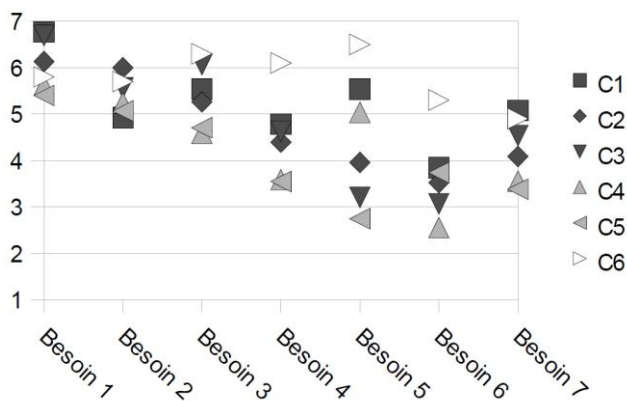
\*\* TEG : Taux d'Etudiants Satisfaits

**Tableau 1 • Usage de Prog&Play dans sept contextes différents et satisfaction globale.**

Notre seconde hypothèse porte sur l'importance de la gestion du temps sur la motivation. Pour le contexte C5, où le jeu a été intégré dans une formation existante, les enseignants ont utilisé le jeu comme un simple outil d'illustration des concepts abordés lors des séances de travaux pratiques. Ce choix a eu pour conséquence de briser la dynamique de *flow* (Csikszentmihalyi, 1991) construite autour de l'approche narrative et de la progression des missions. De plus, les enseignants avaient tendance à presser les étudiants pour qu'ils terminent les exercices dans le temps imparti en leur donnant les solutions des missions. Les étudiants étaient donc dans un processus de suivi de l'enseignant et non dans une démarche de découverte, et d'exploration des possibilités de jeu. Or, dans une session de jeu, les joueurs apprécient de trouver la solution par eux même. Ainsi, l'utilisation de Prog&Play en complément de formations existantes



(C1, C2 et C3) permet d'assurer un meilleur impact du jeu sur la motivation des étudiants. L'intégration la plus satisfaisante pour les étudiants consiste donc à concevoir un enseignement autour du jeu lui même (C6 et C7). Cette intégration demande un investissement certain de la part des enseignants, mais semble être raisonnable puisqu'ils la reconduisent d'année en année.



**Figure 2 • Satisfaction moyenne en fonction du besoin et du contexte d'utilisation.**

La figure 2 présente la satisfaction moyenne pour chaque besoin de la pyramide de Siang et Rao en fonction du contexte. Il apparaît une plus grande dispersion des réponses pour le cinquième besoin (le besoin de connaître et de comprendre). Ce besoin est défini comme la nécessité pour un joueur de découvrir de nouveaux éléments de jeu en vue de les réutiliser dans de nouvelles situations. Dans Prog&Play, ce besoin est satisfait à travers la découverte de nouvelles unités (et de leurs propres caractéristiques) et dans la manière de les commander (construction de programmes). Ceci renforce notre hypothèse selon laquelle le contexte d'utilisation et donc la manière dont le jeu est intégré aux formations influence fortement la motivation. Mais ce cinquième besoin exige que le joueur dispose de temps pour explorer le jeu et semble donc fortement dépendant du temps de jeu alloué au cours de la formation. Plus les étudiants ont passé du temps sur le jeu, plus ce cinquième besoin est satisfait. L'approche par projet du contexte C6 a permis aux étudiants de jouer à Prog&Play autant qu'ils le souhaitent, ils ont donc pu découvrir par eux mêmes les différents éléments du jeu et donc satisfaire ce besoin. Un étudiant du contexte C6 écrit : « *C'est la seule UE où on nous demande de jouer et c'est parce qu'on joue qu'on devient plus fort en programmation. Franchement, j'ai adoré ça* ». Pour les contextes C1 et C4 le jeu a été utilisé lors de 5 séances d'une heure et trente minutes (soit sept heures trente de jeu). Le taux de satisfaction étant correct (autour de 5/7), il semble que ce temps soit suffi-

sant pour que les joueurs obtiennent une bonne vision du jeu. Enfin, les contextes C2, C3 et C5 ont seulement intégré le jeu sur près de quatre heures de formation ce qui semble insuffisant pour découvrir de manière autonome les différentes composantes du jeu.

## 5. Conclusion

Dans cet article, nous avons rapidement décrit Prog&Play, un environnement d'apprentissage s'appuyant sur le jeu. Nous avons étudié les données collectées lorsque le jeu a été introduit dans différents contextes universitaires. Les réponses aux questionnaires étudiants suggèrent une influence du contexte d'enseignement sur la motivation des étudiants : un atelier optionnel de programmation ou un enseignement par projet en complément d'une initiation à la programmation apparaît plus adapté que de remplacer des travaux pratiques traditionnels par le jeu sans autre adaptation par ailleurs. Enfin, nous avons identifié que donner du temps aux étudiants pour découvrir l'univers du jeu et les règles est une des caractéristiques clé pour améliorer la compréhension du jeu et en conséquence, la motivation.

Laisser l'opportunité aux étudiants de réaliser des actions inutiles, redondantes ou incorrectes dans le jeu (Thomas *et al.*, 2011) est fondamental pour captiver l'attention des joueurs et leur permettre de mettre en œuvre par eux-mêmes, les concepts de programmation enseignés. Un étudiant du contexte C6 décrit très bien la motivation induite par l'exploration du jeu : *« Le code de la dernière mission a été réellement très long à établir. Beaucoup, d'idées ont dû être abandonnées et revues dès le début. Au total, des centaines de lignes de code ont été écrites. De nombreuses fois j'ai vu mon armée se faire détruire entièrement. Cependant le fait d'approcher de plus en plus de la victoire à chaque tentative m'a tenu en haleine. Et c'est grâce à cette tension que j'ai fini par la réussir ».*

## Remerciements

Nous remercions chaleureusement les étudiants et les enseignants qui ont participé à cette étude.

## BIBLIOGRAPHIE

(Archambault, 2011)

ARCHAMBAULT P. (2011). Un enseignement de la discipline informatique en Terminale scientifique. *DIDAPRO 4 - Dida&STIC*, p. 205-212.

(Bebras, 2012)

BEBRAS (2012). *Bebras Contest: International Contest on Informatics and Computer Fluency*, <http://www.bebbras.org/en/welcome>, (consulté le 15 octobre 2012).

(Chen et Cheng, 2007)

CHEN W.-K., CHENG Y. C. (2007). Teaching Object-Oriented Programming Laboratory With Computer Game Programming. *IEEE Transactions on Education*, Vol. 50 n°3, p. 197-203.

- (Colobot, 2001)
- COLOBOT (2001). *Colobot*, <http://www.ccebot.com/colobot/index-f.php>, (consulté le 15 octobre 2012).
- (CSC, 2008)
- CSC (2008). ACM, IEEE-CS. *Computer Science Curriculum 2008: An Interim Revision of CS2001*. ACM Press. and IEEE Computer Society Press., New York.
- (Csikszentmihalyi, 1991)
- CSIKSZENTMIHALYI M. (1991). *Flow - The Psychology of optimal Experience*. Harper Perennial.
- (Février *et al.*, 2011)
- FEVRIER L., HIRON M., CHARGUERAUD A (2011). France-IOI : l'apprentissage de l'algorithmique pour tous. *DIDAPRO 4 - Dida&STIC*, p. 213-220.
- (Greitzer *et al.*, 2007)
- GREITZER F. L., KUCHAR O. A., HUSTON K. (2007). Cognitive science implications for enhancing training effectiveness in a serious gaming context. *J. Educ. Resour. Comput.*, Vol. 7 n°3 art. 2.
- (Gun-Tactyx, 2008)
- GUN-TACTYX (2008). *Gun-Tactyx*, <http://apocalyx.sourceforge.net/guntactyx/>, (consulté le 15 octobre 2012).
- (Hartness, 2004)
- HARTNESS K. (2004). Robocode: using games to teach artificial intelligence. *J. of Comput. Sciences in Colleges*, Vol. 19 n°4, p. 287-291.
- (IOI, 2012)
- IOI (2012). *International Olympiad in Informatics*, <http://www.ioinformatics.org/>, (consulté le 15 octobre 2012).
- (Kelleher *et al.*, 2002)
- KELLEHER C., COSGROVE D., CULYBA D., FORLINES C., PRATT J., PAUSCH R. (2002). Alice2: Programming without Syntax Errors. *15th annual symposium on the User Interface Software and Technology*. Disponible sur internet : <http://www.acm.org/uist/archive/adjunct/2002/pdf/demos/p35-kelleher.pdf> (consulté le 15 octobre 2012).
- (Kernel Panic, 2012)
- KERNEL PANIC (2012). *Kernel Panic*, [http://springrts.com/wiki/Kernel\\_Panic](http://springrts.com/wiki/Kernel_Panic), (consulté le 15 octobre 2012).
- (Klopfer et Yoon, 2005)
- KLOPFER E., YOON S. (2005). Developing games and simulations for today and tomorrow's tech savvy youth. *Tech Trends*, Vol. 49 n°3, p. 33-41.
- (Maloney *et al.*, 2004)
- MALONEY J., BURD L., KAFAY Y., RUSK N., SILVERMAN B., RESNICK M. (2004). Scratch: A Sneak Preview. *2nd International Conference on Creating Connecting, and Collaborating through Computing*, p. 104-109.
- (Muratet, 2010)
- MURATET M. (2010). *Conception, réalisation et évaluation d'un jeu sérieux de stratégie temps réel pour l'apprentissage des fondamentaux de la programmation*. Thèse de doctorat, Université Paul Sabatier. Disponible sur internet : [ftp://ftp.irit.fr/IRIT/VORTEX/Muratet\\_these.pdf](ftp://ftp.irit.fr/IRIT/VORTEX/Muratet_these.pdf) (consulté le 15 octobre 2012).
- (Muratet *et al.*, 2011)

MURATET M., TORGUET P., VIALLET F., JESSEL J.-P. (2011). Évaluation d'un jeu sérieux pour l'apprentissage de la programmation. *Revue d'Intelligence Artificielle*, Hermès Science Publications, Vol. 25 n°2, p. 175-202.

(Oblinger, 2004)

OBLINGER D. (2004). The Next Generation of Educational Engagement. *J. of Interactive Media in Education*. Disponible sur internet : <http://www-jime.open.ac.uk/2004/8/oblinger> (consulté le 15 octobre 2012).

(Papastergiou, 2009)

PAPASTERGIOU M. (2009). Digital Game-Based Learning in high school Computer Science education: Impact on educational effectiveness and student motivation. *J. Comput. Educ.*, Vol. 52 n°1, p. 1-12.

(Papert, 1980)

PAPERT S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books: New York.

(Prog&Play, 2012)

PROG&PLAY (2012). *Prog&Play*, <http://www.irit.fr/ProgAndPlay/>, (consulté le 15 octobre 2012).

(PSD, 2012)

PSD (2012). *Programming Skills Development*, <http://pskills.ced.tuc.gr/>, (consulté le 15 octobre 2012).

(Robot Battle, 2009)

ROBOT BATTLE (2009). *Robot Battle*, <http://www.robotbattle.com/>, (consulté le 15 octobre 2012).

(Seehorn *et al.*, 2011)

SEEHORN D., CAREY S., FUSCHETTO B., LEE I., MOIX D., O'GRADY-CUNIFF, D., BOUCHER OWENS B., STEPHENSON C., VERNON A. (2011). *CSTA K-12 Computer Science Standards*. CSTA Standards Task Force.

(Siang et Rao, 2003)

SIANG A. C., RAO R. K. (2003). Theories of learning: a computer game perspective. *Multimedia Software Engineering*, p. 239-245.

(Thomas *et al.*, 2011)

THOMAS P., YESSAD A., LABAT J.-M. (2011). Petri nets and ontologies: tools for the "learning player" assessment in serious games. *Advanced Learning Technologies*, p. 415-419.